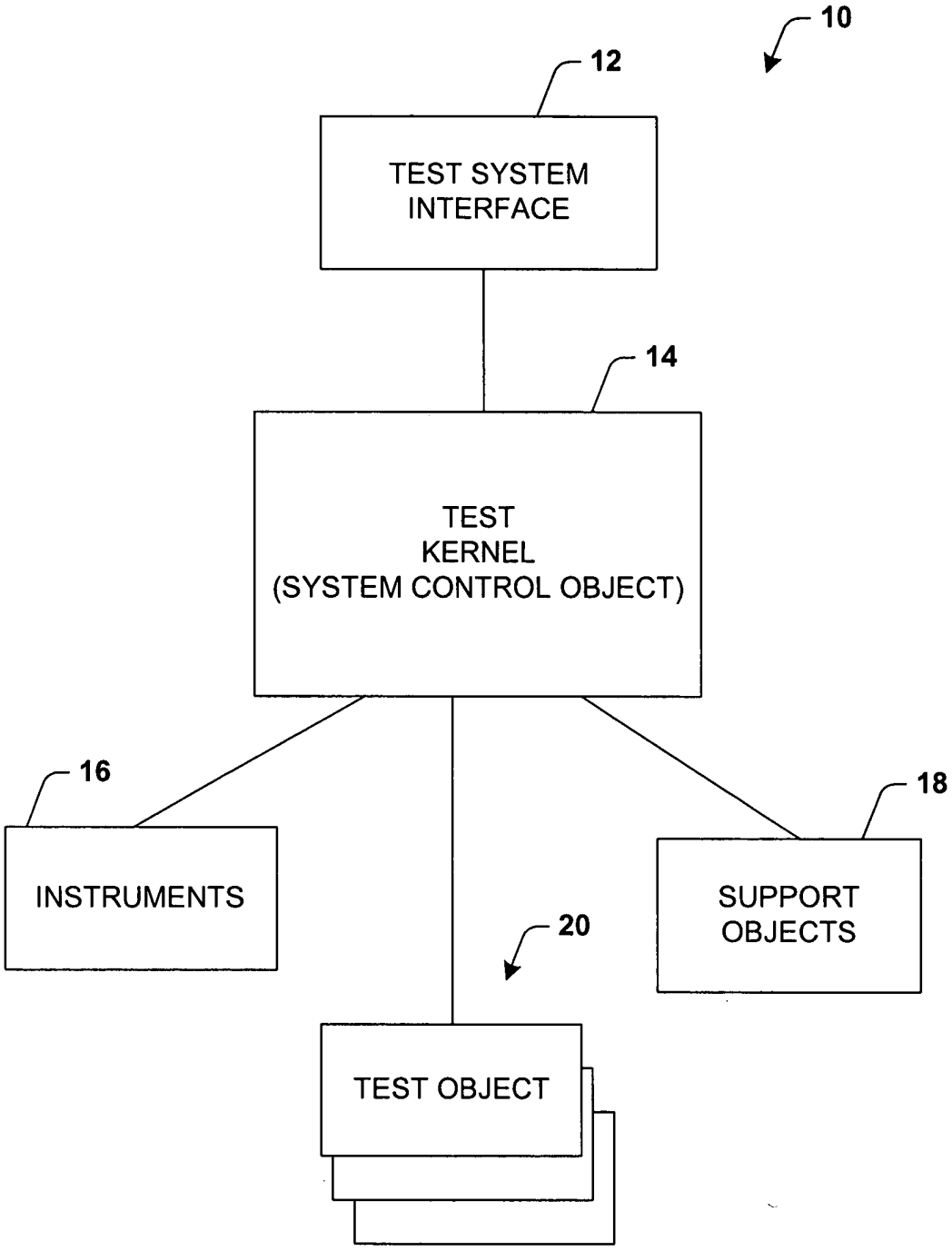
10
12
TEST SYSTEM INTERFACE
14
TEST KERNEL (SYSTEM CONTROL OBJECT)
16
INSTRUMENTS
18
SUPPORT OBJECTS
20
TEST OBJECT

Fig. 1

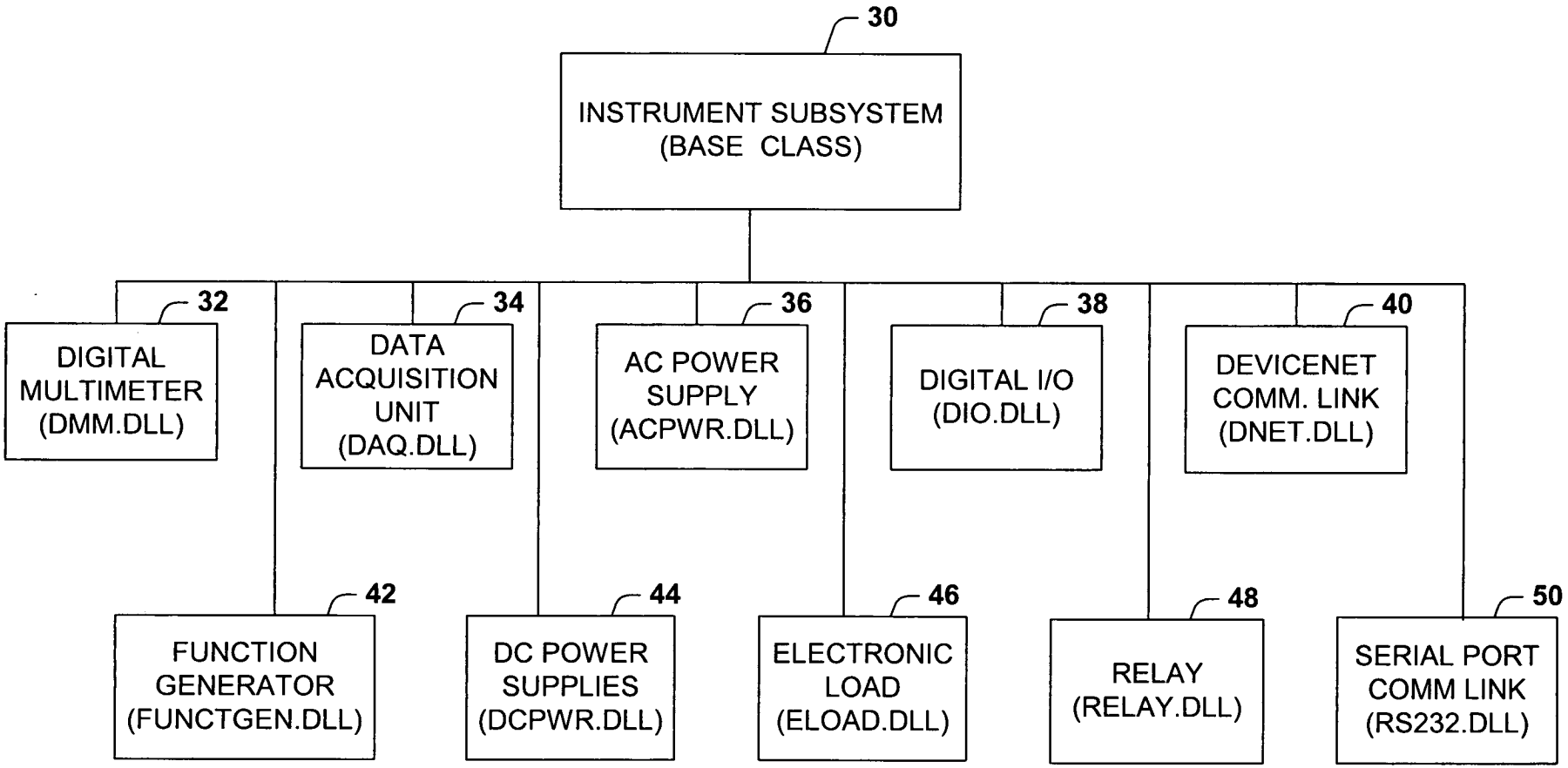
30
INSTRUMENT SUBSYSTEM
(BASE CLASS)
32
DIGITAL
MULTIMETER
(DMM.DLL)
34
DATA
ACQUISITION
UNIT
(DAQ.DLL)
36
AC POWER
SUPPLY
(ACPWR.DLL)
38
DIGITAL I/O
(DIO.DLL)
40
DEVICENET
COMM. LINK
(DNET.DLL)
42
FUNCTION
GENERATOR
(FUNCTGEN.DLL)
44
DC POWER
SUPPLIES
(DCPWR.DLL)
46
ELECTRONIC
LOAD
(ELOAD.DLL)
48
RELAY
(RELAY.DLL)
50
SERIAL PORT
COMM LINK
(RS232.DLL)

Fig. 2a

```
// Dmm.cpp : Defines the entry point for the DLL application.

#include "stdafx.h"
#include "Dmm.h"
#include "instruments.h"                                        <-- 60
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

// Define handler functions
void DmmErrorHandler(INST inst, INT iError);
void DmmSRQHandler(INST id );

// Instrument Control Strings
#define EOI                     "END ALWAYS" // Used to turn on the EOI required by SICL's iread()
#define RESET                   "RESET"
#define QUERY_MODE              "FUNC?"
#define QUERY_RANGE             "RANGE?"
#define QUERY_ARANGE    "ARANGE?"
#define QUERY_FOR_ERROR "ERR?"

BOOL APIENTRY DllMain( HANDLE hModule,
                       DWORD  ul_reason_for_call,
                       LPVOID lpReserved                                )
{
  switch (ul_reason_for_call)
        {
                case DLL_PROCESS_ATTACH:
                case DLL_THREAD_ATTACH:
                case DLL_THREAD_DETACH:
                case DLL_PROCESS_DETACH:
                        break;
  }
  return TRUE;
}

CDmm::CDmm(CHAR *pcDeviceInstance) : CInstr( pcDeviceInstance ) {
        INT iPrimAddress = GetPrimaryAddress(); // Defined in the base class
        INT m_iLiveMode = GetLiveMode();    // Defined in the base class
        m_pCL = (Cleee488 *)NULL;
        CHECK_IF_LIVE(                                          <-- 62
                                m_eMode = DCV;
                                m_eRange = AUTORANGE;
                                return
)
        m_pCL = new Cleee488(iPrimAddress);
        m_pCL->InstallErrorHandler(DmmErrorHandler);
        m_pCL->InstallSRQHandler(DmmSRQHandler);
        m_pCL->SetTimeout(5000);
        m_pCL->Write(EOI, strlen(EOI)); // Turn On The EOI
        GetCurrentDmmState();
}
CDmm::~CDmm() {                                                 <-- 62
        CHECK_IF_LIVE( return )
        delete m_pCL;
}
```

Fig. 2b

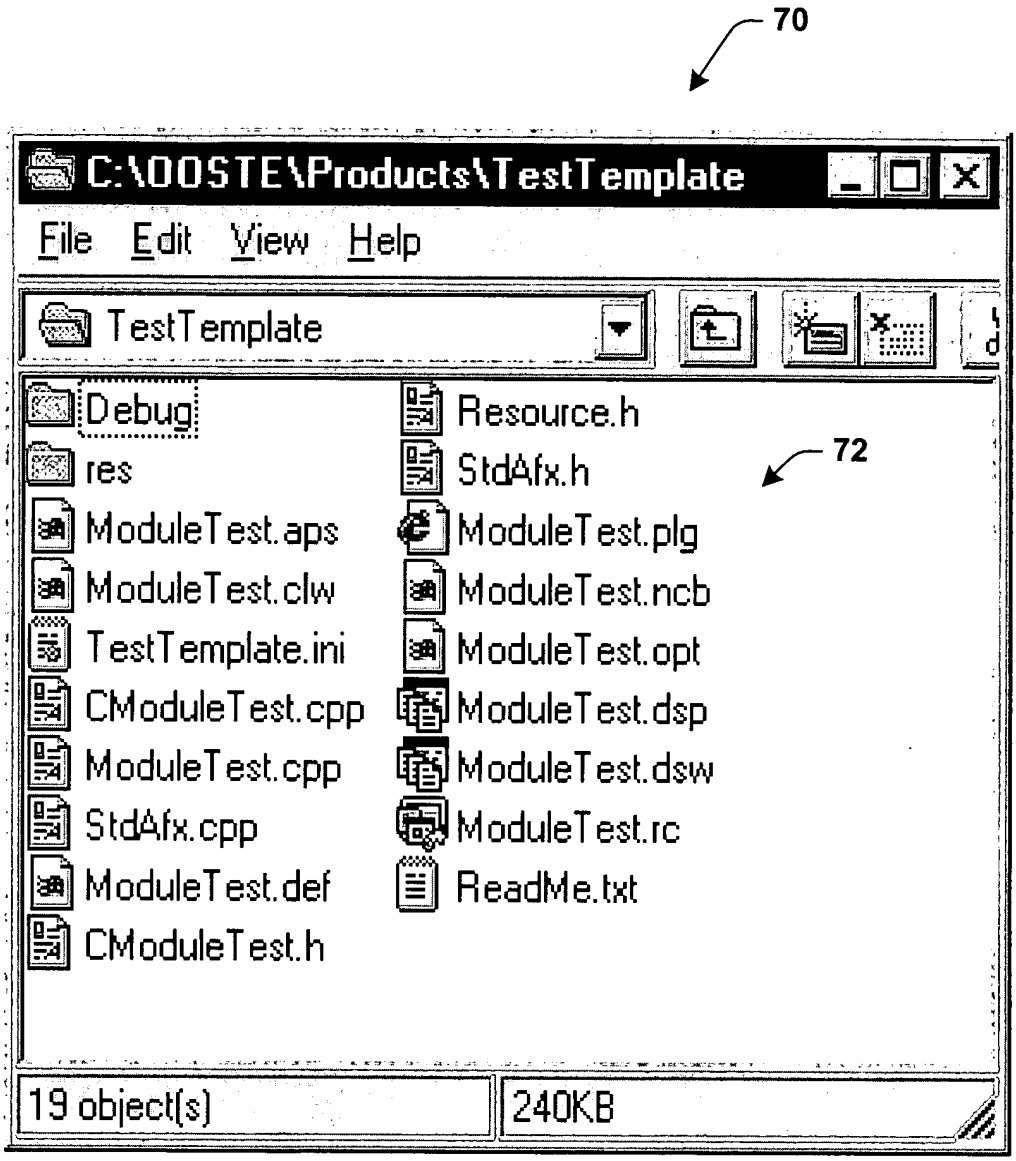
70
C:\OOSTE\Products\TestTemplate
File Edit View Help
TestTemplate
Debug
res
ModuleTest.aps
ModuleTest.clw
TestTemplate.ini
CModuleTest.cpp
ModuleTest.cpp
StdAfx.cpp
ModuleTest.def
CModuleTest.h
Resource.h
StdAfx.h
72
ModuleTest.plg
ModuleTest.ncb
ModuleTest.opt
ModuleTest.dsp
ModuleTest.dsw
ModuleTest.rc
ReadMe.txt
19 object(s)
240KB

Fig. 3

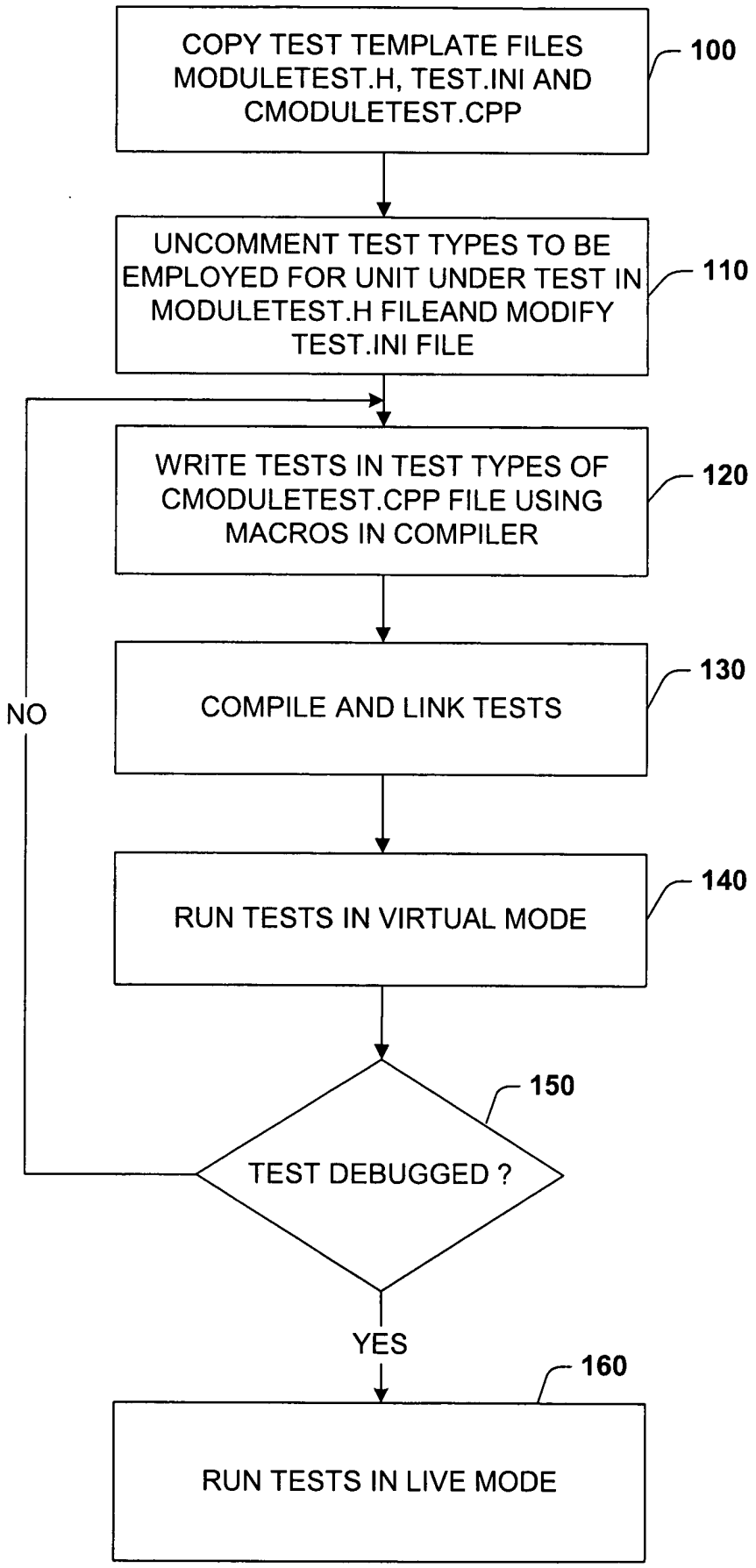
COPY TEST TEMPLATE FILES MODULETEST.H, TEST.INI AND CMODULETEST.CPP
100
UNCOMMENT TEST TYPES TO BE EMPLOYED FOR UNIT UNDER TEST IN MODULETEST.H FILEAND MODIFY TEST.INI FILE
110
WRITE TESTS IN TEST TYPES OF CMODULETEST.CPP FILE USING MACROS IN COMPILER
120
COMPILE AND LINK TESTS
130
NO
RUN TESTS IN VIRTUAL MODE
140
TEST DEBUGGED ?
150
YES
RUN TESTS IN LIVE MODE
160

Fig. 4

```
#ifndef _INC_MODULETEST_H_
#define _INC_MODULETEST_H_

// Required system includes
#include "dmm.h"
#include "eload.h"
#include "dcpwr.h"
#include "acpwr.h"
#include "dio.h"
#include "relay.h"
#include "source.h"
#include "support.h"
#include "Test.h"

class CModuleTest:public CTest
{
public:
        CModuleTest(CRuntimeTestInfo *rRti);
        ~CModuleTest();

        #error Uncomment test level(s) required by your product
        //void Ambient();
        //void Calibration();
        //void Final();
        //void FunctionalModule();
        //void FunctionalSystem();
        //void PreHeatedHot();
        //void Troubleshooting();

        //void InitializeSystem();
        //void CleanupSystem();
        //void GetRequiredInstruments();

private:
        // Add private data members and functions here
protected:
        // Add protected data members and functions here
};
#endif
```

200

202

204

206

Fig. 5a

```
#include "stdafx.h"
#include "Test.h"
#include "CModuleTest.h"
#include "instruments.h"
#include "support.h"

#error Replace SSW#### in the g_pcSoftwareID[] variable declaration below with the correct SSW
number, ie: SSW1234
char g_pcSoftwareID[] = {"SSW####"};
char g_pcSoftwareRevision[] = {"$Revision$"};

// Name: FunctionalModule
// Description:
// Return type: void
// Author: YourNameHere
// Revision History:
//              2/21/2000 12:42:26 PM   Initial Coding          YourInitials
//vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
//void CModuleTest::FunctionalModule()
//{
//}

// Name: FunctionalSystem
// Description:
// Return type: void
// Author: YourNameHere
// Revision History:
//              2/21/2000 12:42:33 PM   Initial Coding          YourInitials
//vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
//void CModuleTest::FunctionalSystem()
//{
//}

// Name: PreHeatedHot
// Description:
// Return type: void
// Author: YourNameHere
// Revision History:
//              2/21/2000 12:42:38 PM   Initial Coding          YourInitials
//vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
//void CModuleTest::PreHeatedHot()
//{
//}
```

210

212

212

212

Fig. 5b

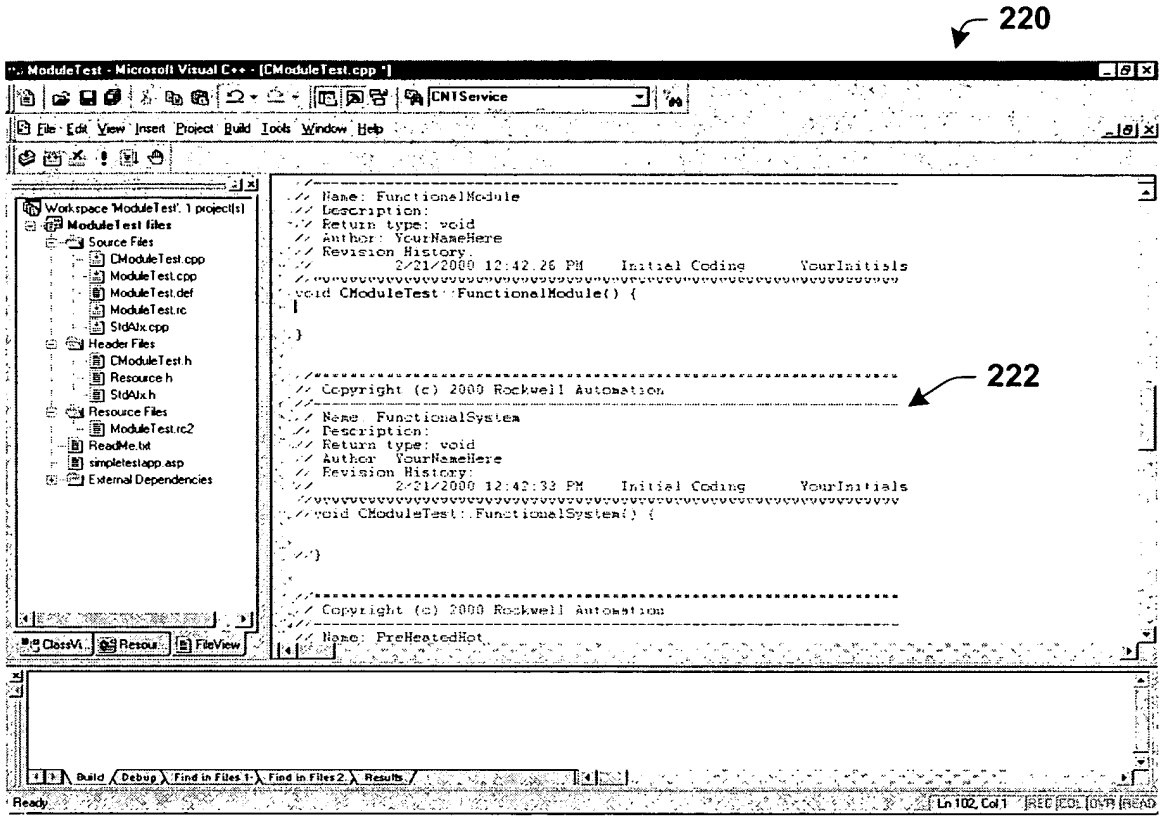
220
222
ModuleTest - Microsoft Visual C++ - [CModuleTest.cpp *]
File Edit View Insert Project Build Tools Window Help
Workspace 'ModuleTest': 1 project(s)
ModuleTest files
Source Files
CModuleTest.cpp
ModuleTest.cpp
ModuleTest.def
ModuleTest.rc
StdAfx.cpp
Header Files
CModuleTest.h
Resource.h
StdAfx.h
Resource Files
ModuleTest.rc2
ReadMe.txt
simpletestapp.asp
External Dependencies
// Name: FunctionalModule
// Description:
// Return type: void
// Author: YourNameHere
// Revision History:
// 2/21/2000 12:42:26 PM Initial Coding YourInitials
void CModuleTest::FunctionalModule() {
}
// Copyright (c) 2000 Rockwell Automation
// Name: FunctionalSystem
// Description:
// Return type: void
// Author: YourNameHere
// Revision History:
// 2/21/2000 12:42:33 PM Initial Coding YourInitials
void CModuleTest::FunctionalSystem() {
}
// Copyright (c) 2000 Rockwell Automation
// Name: PreHeatedHot
Build Debug Find in Files 1 Find in Files 2 Results
Ready
Ln 102, Col 1

Fig. 6a

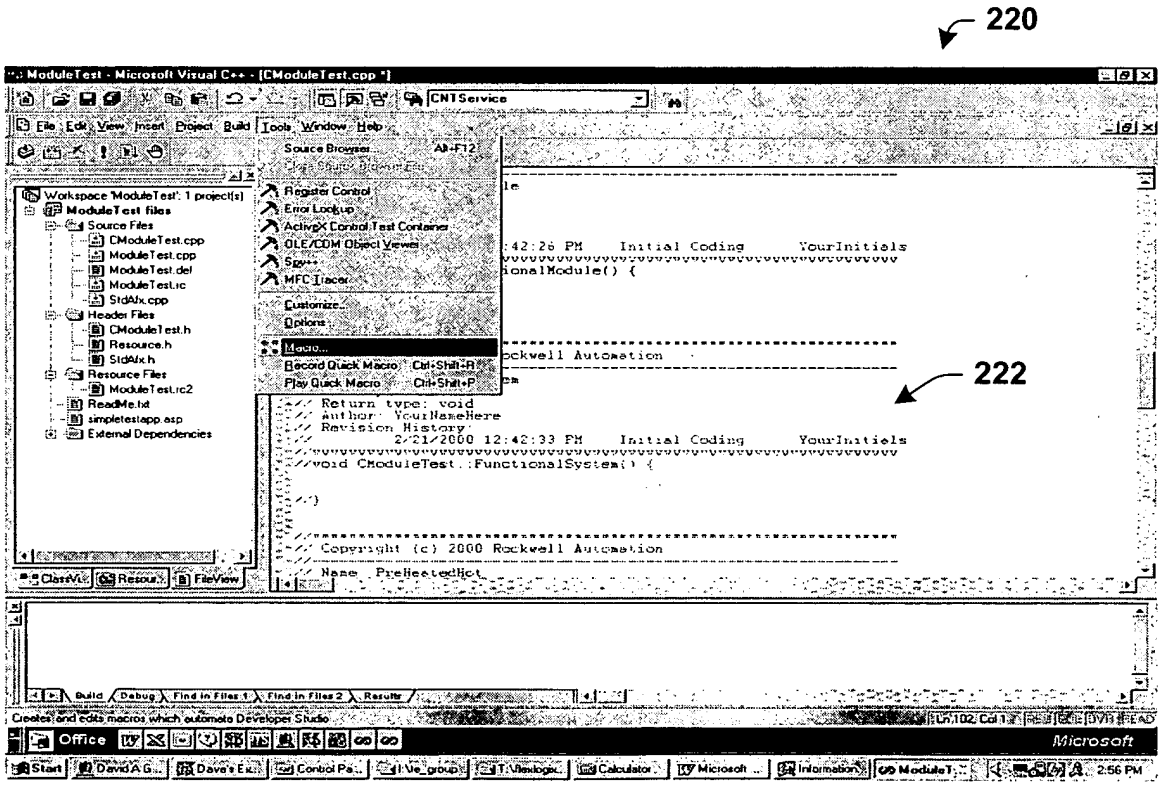
220
222
ModuleTest - Microsoft Visual C++ - [CModuleTest.cpp *]
Source Browser... Alt+F12
Register Control
Error Lookup
ActiveX Control Test Container
OLE/COM Object Viewer
Spy++
MFC Tracer
Customize...
Options...
Macro...
Record Quick Macro Ctrl+Shift+R
Play Quick Macro Ctrl+Shift+P
// Return type: void
// Author: YourNameHere
// Revision History:
// 2/21/2000 12:42:33 PM Initial Coding YourInitials
// Copyright (c) 2000 Rockwell Automation
// Name: PreHeatedHot
Create and edit macros which automate Developer Studio
Office
Microsoft
Start

Fig. 6b

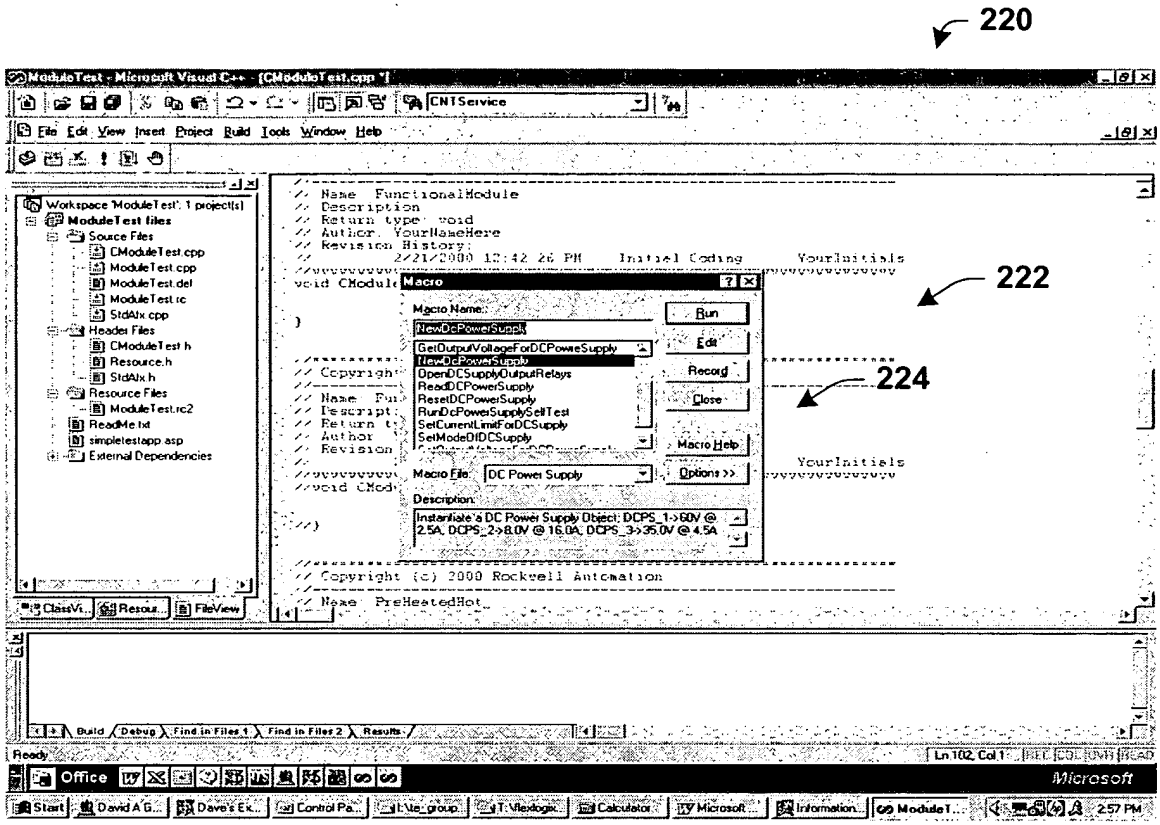
220
222
224
Macro
Macro Name:
NewDcPowerSupply
GetOutputVoltageForDCPowreSupply
NewDcPowerSupply
OpenDCSupplyOutputRelays
ReadDCPowerSupply
ResetDCPowerSupply
RunDcPowerSupplySelfTest
SetCurrentLimitForDCSupply
SetModeOfDCSupply
Run
Edit
Record
Close
Macro Help
Options >>
Macro File:
DC Power Supply
Description:
Instantiate a DC Power Supply Object: DCPS_1->60V @ 2.5A, DCPS_2->8.0V @ 16.0A, DCPS_3->35.0V @ 4.5A
// Name: FunctionalModule
// Description
// Return type: void
// Author: YourNameHere
// Revision History:
// 2/21/2000 12:42:26 PM Initial Coding YourInitials
// Copyright (c) 2000 Rockwell Automation
// Name: PreHeatedHot

Fig. 6c

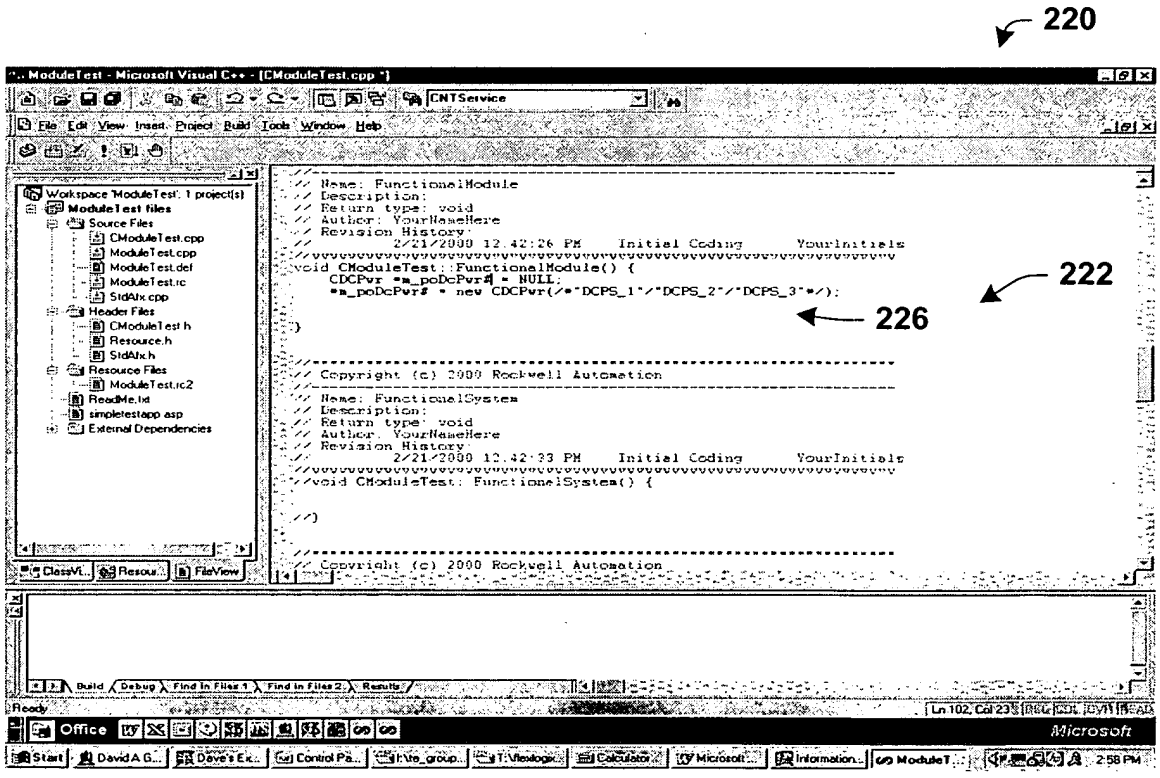
220
222
226
// Name: FunctionalModule
// Description:
// Return type: void
// Author: YourNameHere
// Revision History:
// 2/21/2000 12:42:26 PM Initial Coding YourInitials
void CModuleTest::FunctionalModule() {
CDCPwr *m_poDcPwr# = NULL;
m_poDcPwr# = new CDCPwr(/"DCPS_1"/"DCPS_2"/"DCPS_3"*/);
}
// Copyright (c) 2000 Rockwell Automation
// Name: FunctionalSystem
// Description:
// Return type: void
// Author: YourNameHere
// Revision History:
// 2/21/2000 12:42:33 PM Initial Coding YourInitials
//void CModuleTest::FunctionalSystem() {
// Copyright (c) 2000 Rockwell Automation

Fig. 6d

```
#include "stdafx.h"
#ifndef _INC_MODULETEST_H_
#define _INC_MODULETEST_H_

// Required system includes
#include "dmm.h"
#include "eload.h"
#include "dcpwr.h"
#include "acpwr.h"
#include "dio.h"
#include "relay.h"
#include "source.h"
#include "support.h"
#include "Test.h"

class CModuleTest:public CTest {
public:
      CModuleTest(CRuntimeTestInfo *rRti);
      ~CModuleTest();

//    #error Uncomment test level(s) required by your product
      //void Ambient();
      //void Calibration();
      //void Final();
      void FunctionalModule();
      //void FunctionalSystem();
      //void PreHeatedHot();
      //void Troubleshooting();
      void InitializeSystem();
//    void CleanupSystem();
//    void GetRequiredInstruments();

private:

//    #error Uncomment the instruments required by your test
      CDmm  *m_pDmm;
//    CDaq  *m_pDaq;
//    Source      *m_pSource;
      CDCPwr      *m_pDc1;
//    CDCPwr      *m_pDc2;
      CDCPwr      *m_pDc3;
//    CACPwr      *m_pAc;
      CRelay      *m_pRelays;
      CDio  *m_pDio;
//    CELoad      *m_pLoad;

protected:
};
#endif

extern "C" __declspec(dllexport) CModuleTest* CreateObject(CRuntimeTestInfo
*rRti);
extern "C" __declspec(dllexport) void GetSoftwareId( char *pcSwIdBuf, int iSizeOfSwIdBuf, float *fSwRev );
```

250

252

254

Fig. 7a

270

```
#include "stdafx.h"
#ifndef _INC_ void CModuleTest::FunctionalModule() {

      // Define all instruments used in this test (Typically this
      // would be done in this classes constructor, but is placed here
      // for clarity

      CDCPwr *m_poDcPwr1 = NULL;
      CRelay *m_poRelays = NULL;
      CDmm *m_poDmm = NULL;

      // Instantiate the following instrument; DC Power Supply, Relay,
      // and DMM. Again, this would typically be done within the
      // classes constructor but is added here for clarity
      m_poDcPwr1 = new CDCPwr("DCPS_1");
      m_poRelays  = new CRelay("RELAYS");
      m_poDmm = CDmm("DMM_1");

      // Close Power Relay's 0 & 1. This illustrates one way the
      // DC power supply could be connected to the unit under test.
      m_poRelays->Close(PR, 0);
      m_poRelays->Close(PR, 1);

      // Now, close Analog Relays 0 & 1. This illustrates how the DMM
      // could be connected to the power supply connector on the unit
      // under test.
      m_poRelays->Close(AR, 0);
      m_poRelays->Close(AR, 1);

      // Now setup the DMM to operate in DCV mode and AUTORANGE
      m_poDmm->SetMode( DCV );
      m_poDmm->SetRange( AUTORANGE );

      // Display the current test step on the screen for the operator
      DisplayTestStep( "Power Supply Test");

      // Set the DC power supply to output 24Vdc with a max. current of
      // 1.0 Amps
      m_poDcPwr1->SetOutputValue( 24.0 );
      m_poDcPwr1->SetCurrentLimit( 1.0 );

      // Perform a reading from the DMM
      double dReading = m_poDmm->Read();

      // Check that the reading performed above is between 25.0Vdc
      // & 23.0 Vdc. If this fails, the test will terminate and the
      // error logged
      VerifyAndLogError("Verify Unit 24VDC Input", dReading, 25.0,
                              23.0, "VDC");


      // Reset all of the Analog Relays in the system
      m_poRelays->Reset(AR);
```

Fig. 7b

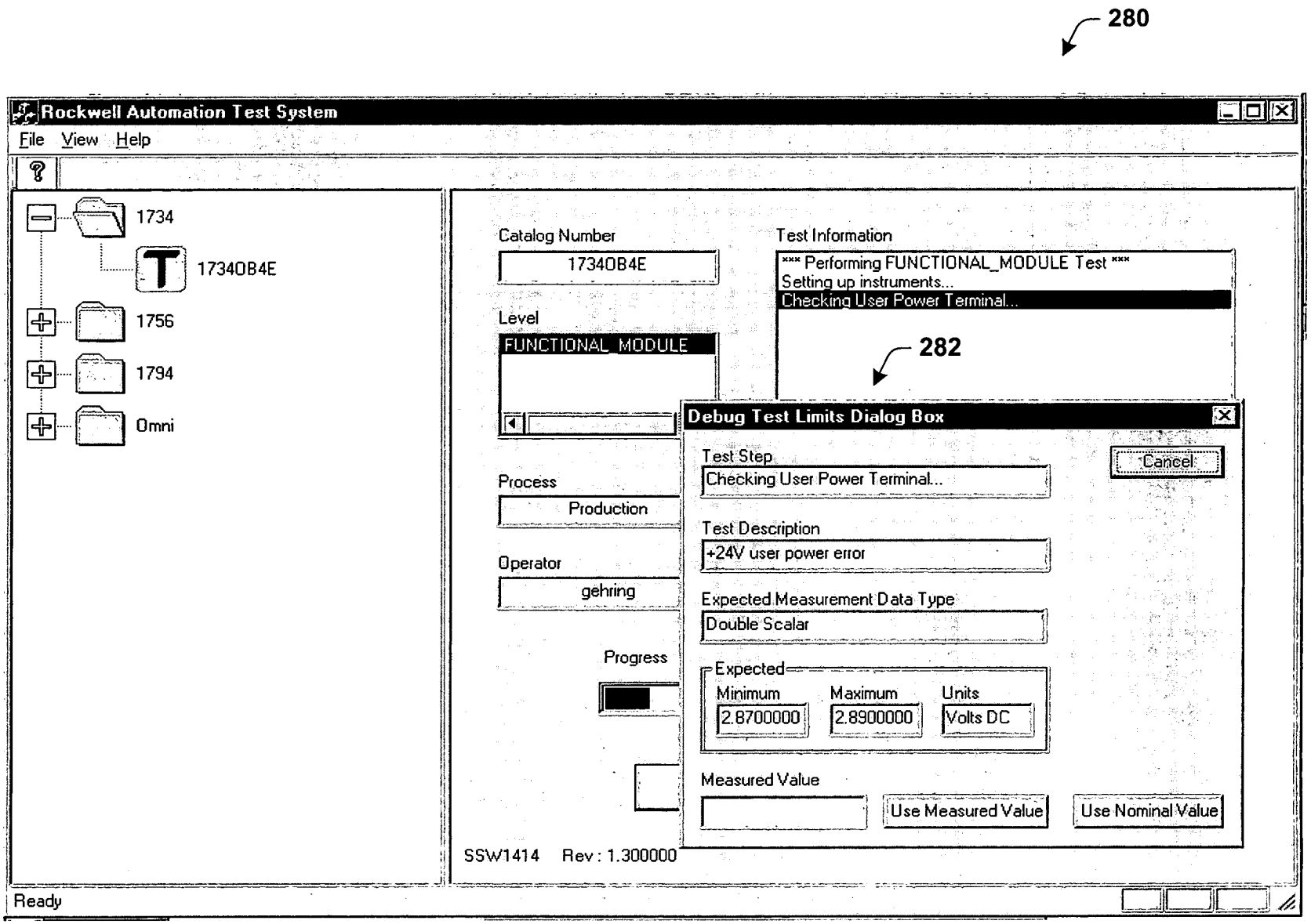
280
Rockwell Automation Test System
File View Help
1734
1734OB4E
1756
1794
Omni
Catalog Number
1734OB4E
Test Information
*** Performing FUNCTIONAL_MODULE Test ***
Setting up instruments...
Checking User Power Terminal...
Level
FUNCTIONAL_MODULE
282
Debug Test Limits Dialog Box
Test Step
Checking User Power Terminal...
Cancel
Process
Production
Test Description
+24V user power error
Operator
gehring
Expected Measurement Data Type
Double Scalar
Progress
Expected
Minimum
2.8700000
Maximum
2.8900000
Units
Volts DC
Measured Value
Use Measured Value
Use Nominal Value
SSW1414 Rev : 1.300000
Ready

Fig. 8

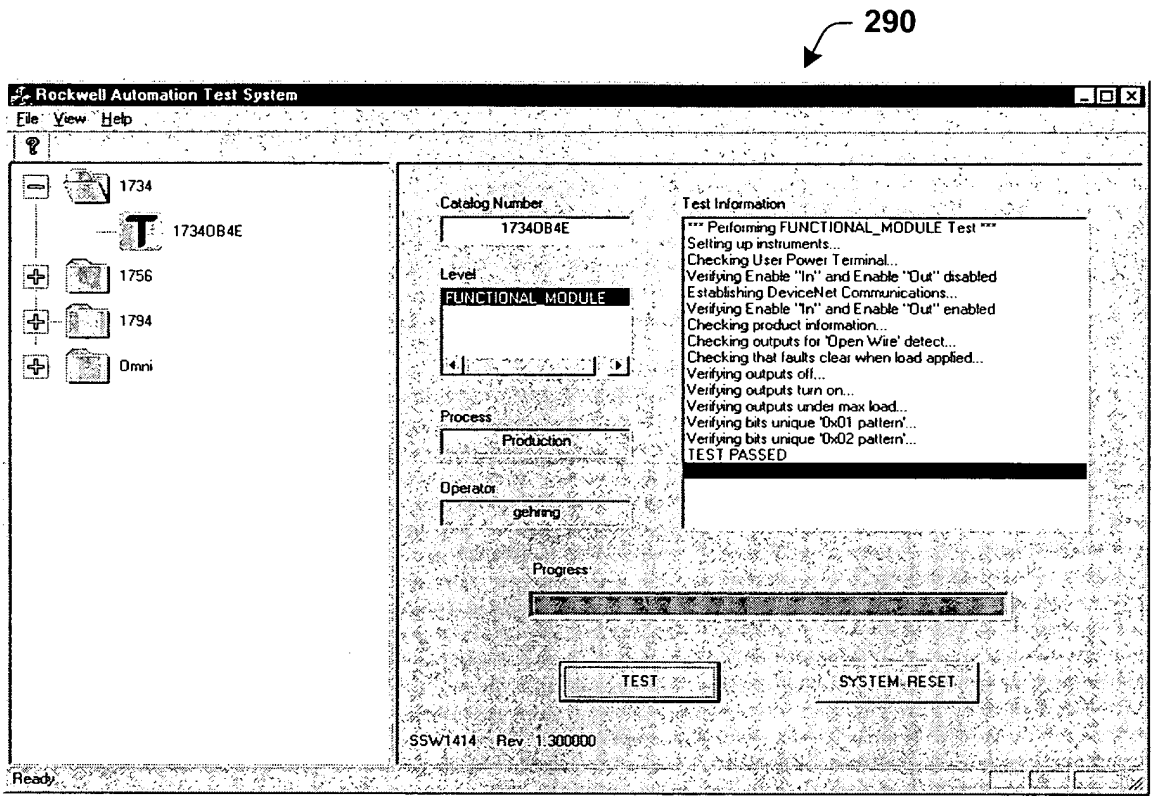
290
Rockwell Automation Test System
File View Help
1734
1734OB4E
1756
1794
Omni
Catalog Number
1734OB4E
Level
FUNCTIONAL_MODULE
Process
Production
Operator
gehring
Test Information
*** Performing FUNCTIONAL_MODULE Test ***
Setting up instruments...
Checking User Power Terminal...
Verifying Enable "In" and Enable "Out" disabled
Establishing DeviceNet Communications...
Verifying Enable "In" and Enable "Out" enabled
Checking product information...
Checking outputs for 'Open Wire' detect...
Checking that faults clear when load applied...
Verifying outputs off...
Verifying outputs turn on...
Verifying outputs under max load...
Verifying bits unique '0x01 pattern'...
Verifying bits unique '0x02 pattern'...
TEST PASSED
Progress
TEST
SYSTEM RESET
SSW1414 Rev 1.300000
Ready

Fig. 9a

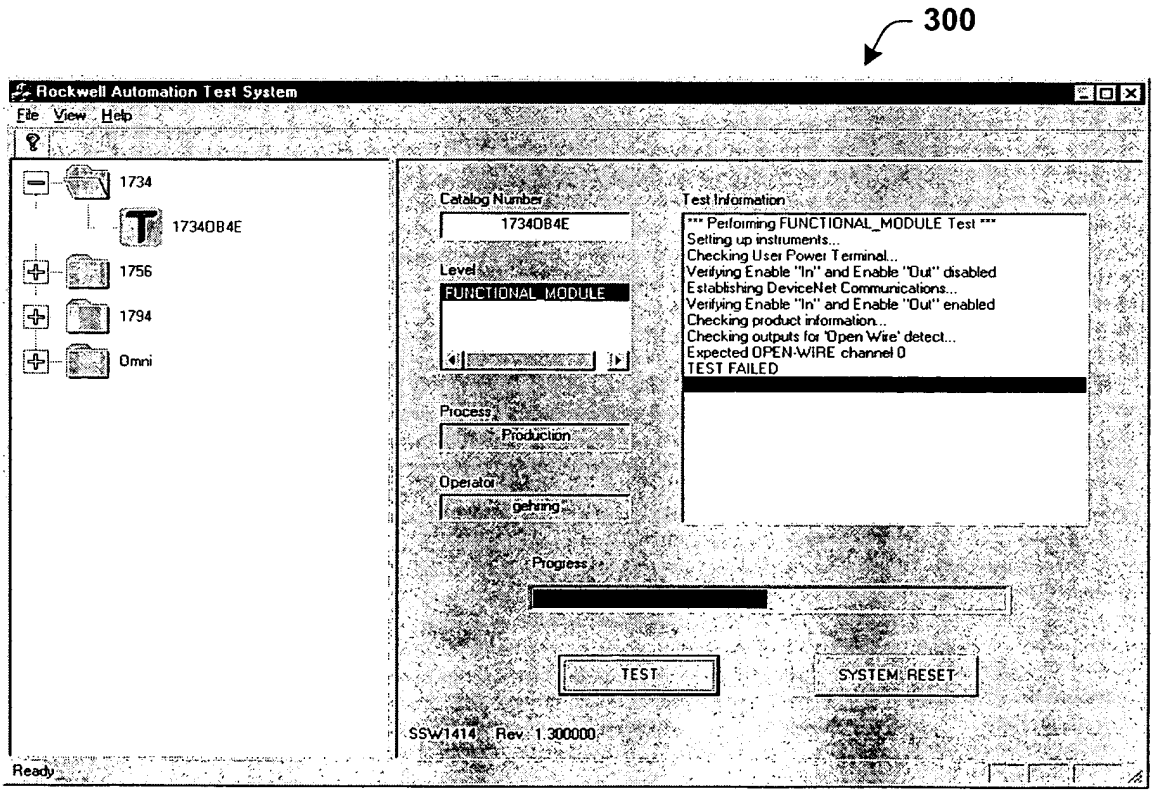
300
Rockwell Automation Test System
File View Help
1734
1734OB4E
1756
1794
Omni
Catalog Number
1734OB4E
Level
FUNCTIONAL_MODULE
Process
Production
Operator
gehring
Test Information
*** Performing FUNCTIONAL_MODULE Test ***
Setting up instruments...
Checking User Power Terminal...
Verifying Enable "In" and Enable "Out" disabled
Establishing DeviceNet Communications...
Verifying Enable "In" and Enable "Out" enabled
Checking product information...
Checking outputs for 'Open Wire' detect...
Expected OPEN-WIRE channel 0
TEST FAILED
Progress
TEST
SYSTEM RESET
SSW1414 Rev 1.300000
Ready

Fig. 9b

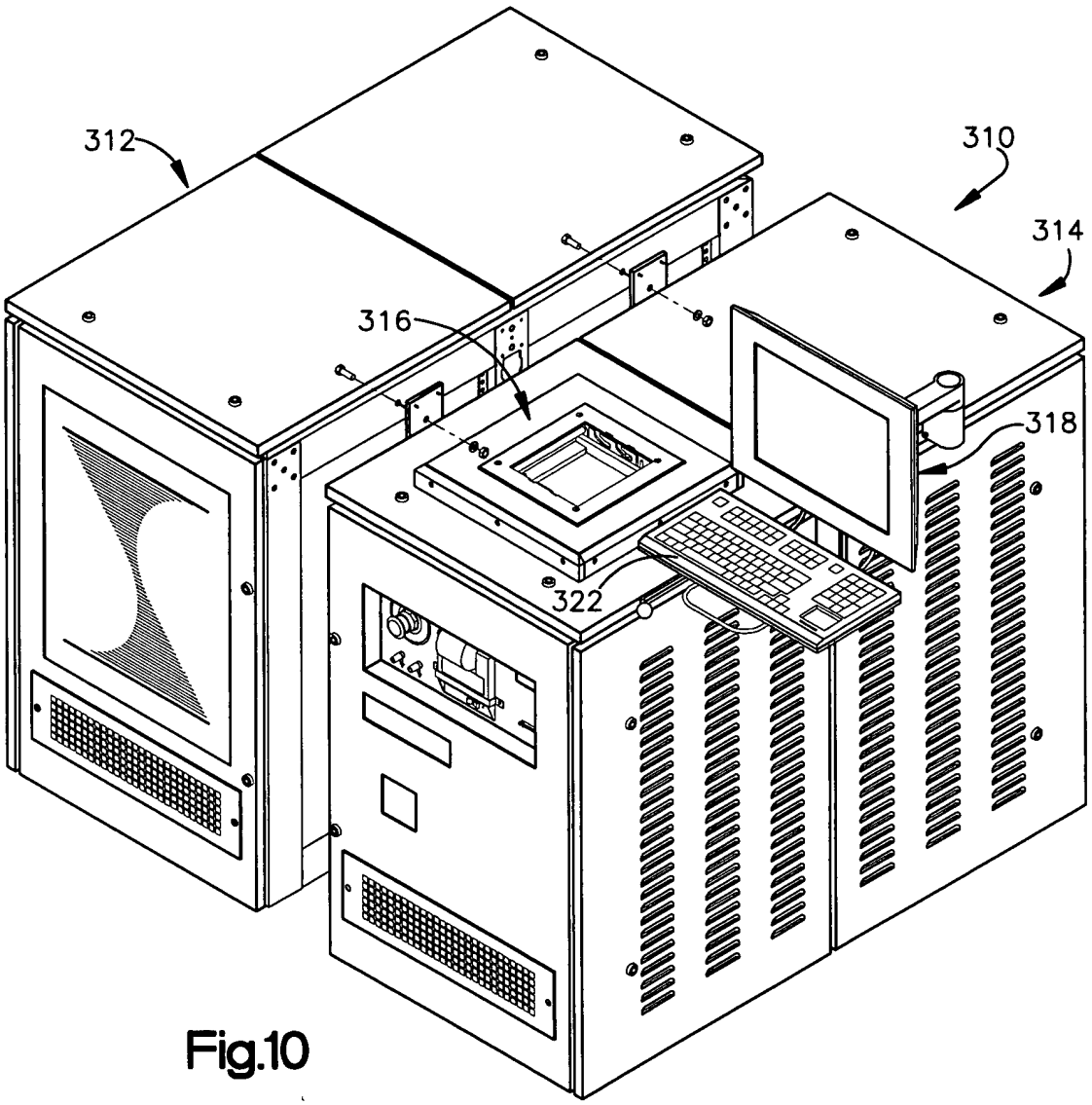

Fig.10

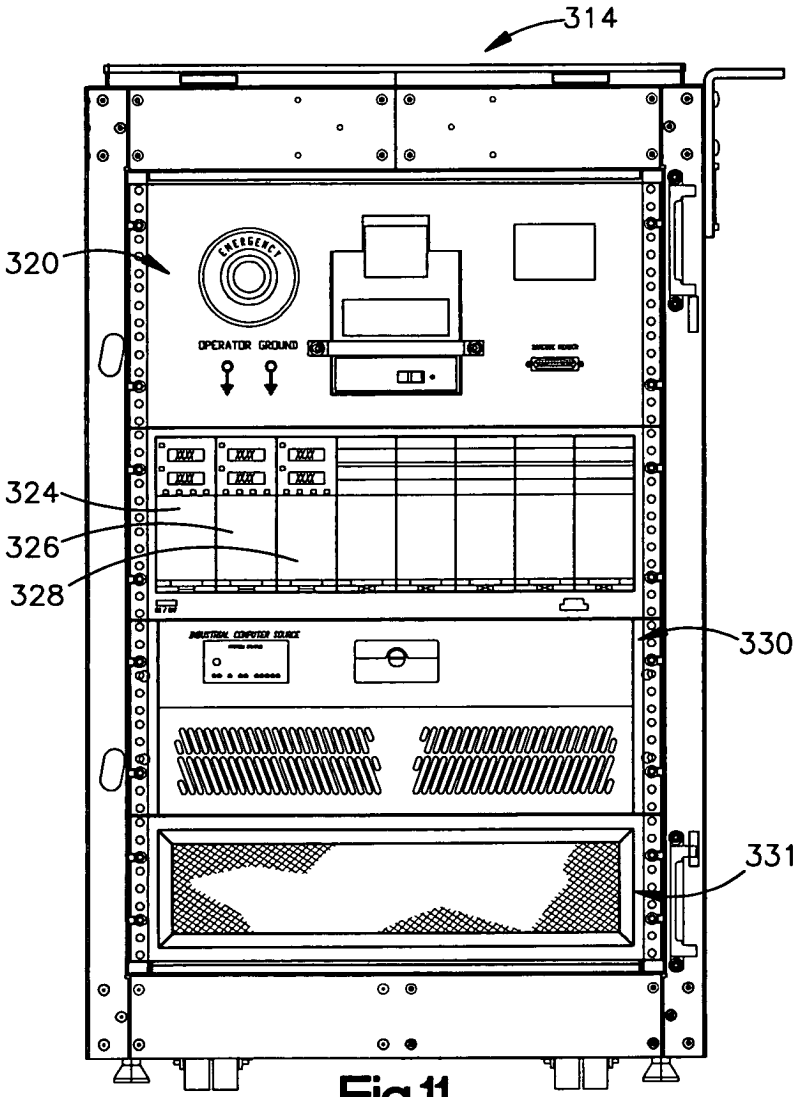
314
320
EMERGENCY
OPERATOR GROUND
324
326
328
330
331

Fig.11

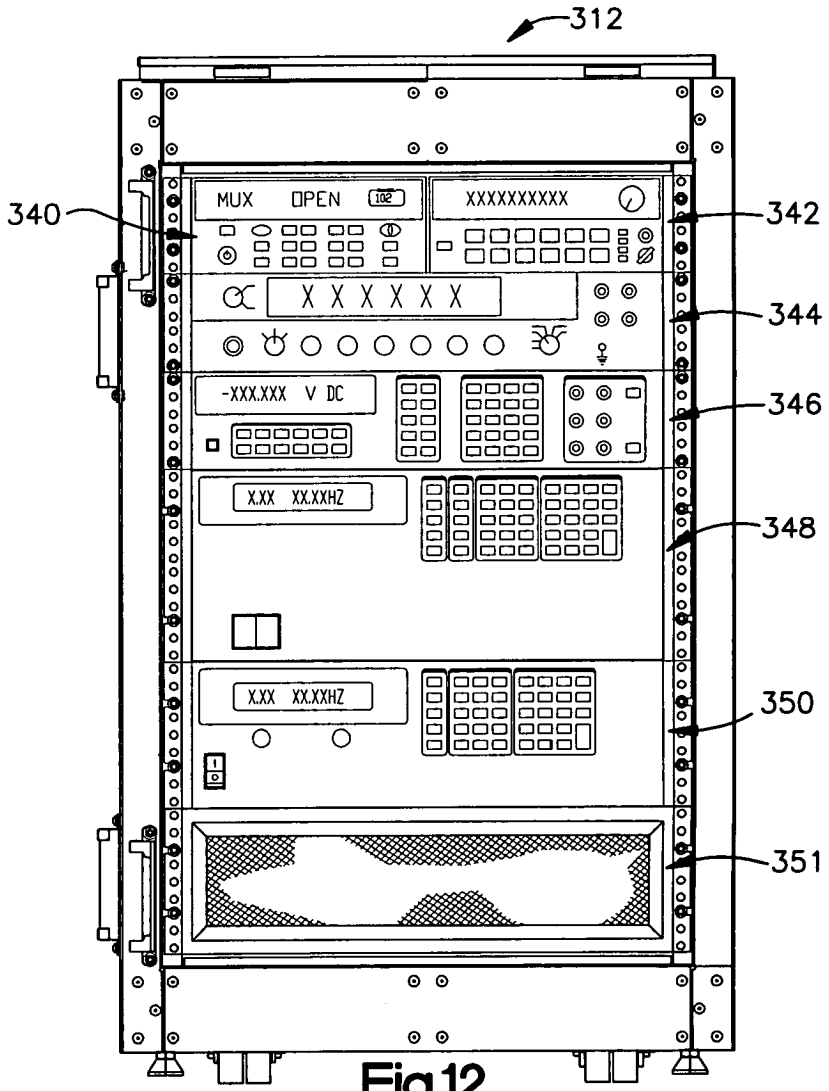
312
340
MUX OPEN
XXXXXXXXXX
342
X X X X X X
344
-XXX.XXX V DC
346
X.XX XX.XXHZ
348
X.XX XX.XXHZ
350
351

Fig.12